Chapter 8
Cloud Computing: A Mobile Context-Awareness Perspective

Nayyab Zia Naqvi, Davy Preuveneers, and Yolande Berbers

Abstract Mobile devices and their applications take advantage of cloud computing not only to overcome their limitations in computing, memory and storage resources but also to easily scale up to a large number of users. Many common mobile applications we use every day, such as e-mail, news, social networking and games, already rely on the cloud paradigm due to these reasons. Another trend of the last decade is the evolution towards smart applications that take into account the user's context, such as their whereabouts and ongoing activities, to adapt the behaviour of the applications according to the current situations. With context management and inferences becoming fairly complex processes, cloud services can offer an edge by taking over the most computationally intensive context management tasks and have these tasks carried out only once for multiple users. However, as a lot of this contextual information is being sensed and captured by the mobile devices themselves, it is not always clear whether it makes sense to upload all the raw sensed data to the cloud and have it completely processed there. These trade-offs depend on the amount of data being communicated and the kind of processing that is required. In this chapter, we analyse the challenges to develop context-aware cloud services for mobile applications and present a conceptual architecture for context processing and service provisioning in the Platform-as-a-Service layer. We highlight various architectural viewpoints of the presented federated context-processing middleware and round up this chapter with a research road map for the future.

Keywords Context-aware perspective • Mobile computing • Social networking • Context management

N.Z. Naqvi (✉) • D. Preuveneers • Y. Berbers
iMinds-DistriNet, Department of Computer Science, KU Leuven, 3001 Leuven, Belgium
e-mail: nayyab.naqvi@cs.kuleuven.be; davy.preuveneers@cs.kuleuven.be; yolande.berbers@cs.kuleuven.be

Z. Mahmood (ed.), *Cloud Computing: Methods and Practical Approaches*,
Computer Communications and Networks, DOI 10.1007/978-1-4471-5107-4_8,

8.1 Introduction

The proliferation of smart applications is giving rise to a rapidly evolving subject area known as mobile cloud computing. Advances in mobility and wireless communications have increased the use of smart mobile applications. Many of the smart applications are now looking at the cloud-computing paradigm as an enabler to exploit its raw computing power, memory and storage resources to overcome the resource limitations of mobile devices [1]. As such, cloud services have not only changed the way in which we run these applications but also how we develop them. Single monolithic applications for the mobile have now evolved towards a distributed loosely coupled composition of services where some of the service components are still running on the mobile but where others are running in the cloud. Such an approach requires an upfront investment in the methodology of how to develop mobile cloud applications as it imposes new integration and interoperability requirements.

Furthermore, as many of the applications try to become smarter, they aim to understand and anticipate the situation of their users by alerting them in time, or adapting their behaviour according to the new context, and this without any intervention of their users [2]. As such, context management becomes an integral part of the mobile cloud applications, with some of these tasks running on the mobile (e.g. context sensing and simple processing) and other computationally more expensive context management tasks being outsourced and taking place remotely on the cloud infrastructure (e.g. pattern detection, reasoning and learning). For some tasks, the decision of what to do where is not always so clear-cut. This decision depends on various overhead trade-offs, including the quality of service (QoS) and the quality of experience (QoE). These trade-offs involve computation, communication, storage, energy consumption, availability, connectivity, as well as user preferences with respect to privacy and security for sensitive data.

The mobile cloud-computing paradigm from a context-aware perspective can be regarded as a research track that aims to find effective ways to make cloud services, being utilized in mobile applications, aware of the context of their customers and applications. To ease the development of context-aware applications, it is paramount to decouple the application logic and its adaptation logic from context acquisition and inference, and at the same time, it is equally important to provide efficient and effective ways to utilize the acquired context. That is why writing scalable and robust smart applications for mobile devices in the cloud environment not only involves being mindful of the architecture of both the cloud services and the management of the context information [3] but also about the communication and coordination between all the parties involved that drive the smart adaptive behaviour of the application.

In this chapter, we provide an overview of how context awareness aims to make mobile applications smart and discuss challenges of how to effectively utilize the cloud for this particular type of applications. We also discuss architectural blueprints for the design of a cloud-based context-processing middleware that can adapt dynamically and off-load its components to the cloud for smart mobile applications to save resources of the mobile devices. The chapter has been organized into seven sections. In Sect. 8.2, we introduce the notion of context and some definitions and the role of context awareness in mobile applications. We depict various major building blocks for context-processing frameworks and discuss various well-known systems found in the literature. In Sect. 8.3, we further explore the role of the cloud and discuss the main challenges of outsourcing context management to the cloud in an efficient and effective manner. Architectural considerations for the context-aware applications are investigated in Sect. 8.4. The next section discusses the implementation flow of the identified services and an evaluation strategy, while in Sect. 8.6 we present some guidelines and a research road map for the future ending with a conclusion focusing on the significance of leveraging the cloud-computing paradigm for context-aware smart applications in Sect. 8.7.

8.2 Overview of Context-Aware Mobile Applications and Architectural Trends

In this section, we discuss the role of context awareness for intelligent behaviour of mobile applications and highlight the existing context-processing frameworks and their architectures.

8.2.1 Defining Context and the Role of Context Awareness

Context awareness is an area of research where applications aim to take into account information about their user and his situation and change the behaviour of the applications accordingly. Context awareness is an intrinsic characteristic of intelligent environments [4–6]. Context was first introduced by Schilit and Theimer [7] in 1994, and its definition has evolved ever since. It was initially referred to as location, identities of nearby people, objects and changes to these objects. Chen et al. [8] proposed a definition of the context as

> Context is the set of environmental states and settings that either determines an application's behaviour or in which an application event occurs and is interesting to the user.

Dey [4] elaborated in his most cited definition for context on the interaction of an application with its user, his environment and situation of the resources involved:

> Context is any information that can be used to characterize the situation of entities (i.e. whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves. Context is typically the location, identity and state of people, groups and computational and physical objects.

Chaari [9] defined context in terms of dynamic application behaviour on the basis of user's varying context as follows:

> Context is a set of the external parameters that can influence the behaviour of an application by defining new views on its data and its available services. These parameters may be dynamic and may change during the execution.

There are many more definitions for context, and it is impossible to list them all. However, it is clear that context information is dependent on individual users, systems and circumstances, as one piece of information might be considered as context information in one scenario but not in another one. It all depends on whether the information is relevant for the application to make well-informed autonomous decisions or manifest autonomous behaviour. In this chapter, we will mainly consider all context information that can be used to improve the behaviour of the mobile application services in a particular situation while at the same time optimize the context management itself by leveraging cloud-computing capabilities.

8.2.2 Context Awareness in Mobile Applications

In mobile applications, context awareness can be regarded as the dynamic adaptation of these applications according to the continuously changing context of the user, the communication network, the physical environment and the device itself [10, 11]. For instance, if a user wants to download and play a video in his mobile device, a context-aware video playing application on the mobile device may automatically select the appropriate variant according to its available bandwidth, memory and battery power and display size of the device and if necessary off-load the transcoding of the original video to this variant to the cloud. This way, the mobile device gets a suitable video stream with respect to its own capabilities and resource availability.

Context awareness requires an effective approach to cater for getting the right information at the right time to make the correct decisions for changing or optimizing its behaviour. Context awareness usually involves three basic steps [7]:

- *Discovery*: This involves the identification of entities that are relevant for the application's tasks. In the first step, a context-aware application has to discover and explore its environment in order to get information to work with. In the above example, it not only means contextual information about resources on its own device but also about other networks, computational facilities and displays in the neighbourhood.
- *Selection*: A context-aware application has to filter the information that was discovered according to its specific needs. Information about displays that are located on the floor below is irrelevant. This selection process can be fairly complex as it may require complex filtering techniques to decide which sensor or device is offering relevant information.

- *Use*: The application uses the discovered and selected information to make the right decision or change its configuration or behaviour. We have identified several high-level ways of how context can be utilized:
  - Contextual information display: The context can be presented to the user. A simple example is finding the closest Mexican restaurant to your current location.
  - Contextual augmentation: Augmentation [10] will associate the metadata about the captured context, for example, showing on a map all the shops that sell a particular item, with nearby parking facilities and that will still be open for at least half an hour by the time you reach the shop.
  - Contextual mediation: This use case focuses on getting relevant context information to the entities that need it (but they might not be able to sense it themselves) [12, 13]. For example, remote applications might optimize their services or functionality to best meet the needs and limits arising from the local and evolving context of interaction.
  - Context-aware configuration: This use of context defines the rules for the actions to be performed in a given context, e.g. choosing a nearby printer or relocating the display to another screen.
  - Context-triggered actions: This context-aware behaviour will initiate certain steps when certain contextual conditions are met, e.g. turning on the silent mode of the mobile and hiding the notifications if the user is in a meeting.

In the following section, we discuss various frameworks and systems that have been used for the development prototype context-aware applications.

8.2.3 Frameworks and Architectures for Context-Aware Applications

Baldauf [14] discusses in his survey a number of context-processing systems and their architectures, including the Context-Awareness Sub-Structure (CASS) [15], Service-Oriented Context-Aware Middleware (SOCAM) [16] based on RMI, Context Broker Architecture (CoBrA) [17], Hydrogen [18] and the Context Toolkit [19], with the latter probably being the most famous one. Every system has its own processing approach, ways to describe the relevant context information in a specific scenario, as well as its own context communication and sharing protocols.

Some of the above context-processing systems are network based, while others have their components tightly coupled. However, we can identify the basic context-processing tasks to achieve context awareness, as follows:

- Context sensing and fetching
- Context filtering and aggregation
- Context storage
- Context reasoning
- Context use

In general, the architectural design for context-aware applications took off with a completely centralized approach dealing with the context of a single user and a single application, with the limited context management capabilities completely embedded in the application. Later on, context management was abstracted from the application logic with a dedicated context-aware processing server, allowing better reuse when running multiple local applications atop. Gradually, context management evolved into a distributed architecture, where context-aware applications leveraged the same remote centralized context-processing server, allowing for multiple remote users to consume personalized services.

However, there has been relatively little advancement in context-aware computing over the past 5 years on how to make context management itself more effective and efficient through distribution of the tasks involved and the overall workload. The work presented in this chapter attempts to enable a new phase of context-aware mobile application development using the cloud-computing paradigm for better scalability and reusability of context management by implementing loosely coupled context-supporting components to achieve the aforementioned context-processing tasks.

8.2.4 A Service-Oriented Approach for Context-Aware Mobile Applications

As discussed in the previous section, a number of architectures supporting context-aware applications have been developed, but little attention is paid to the special requirements and constraints of mobile devices. Many of the state-of-the-art centralized deployment models, application-specific communication protocols and context data representations are not easily applicable in a cloud environment, where sharing of the context between interlinked services raises new challenges of context data representation, storage, processing and distribution in a service-oriented way. Hong et al. [20] have highlighted the advantages and the need of a service-oriented infrastructure abstraction for robust context-aware systems.

A cloud-computing paradigm is an ecosystem built on top of web services [21]. In order to manage its processing and composition in a flexible manner, the supporting components of context processing need to be provisioned as loosely coupled web services. This realization in the form of web services requires representation and aggregation of context information in a structured and standard format. In current context-aware systems, XML-based languages are widely used for modelling and representing the context information. The Pervasive Profile Description Language (PPDL) is an XML-based language to represent situational context [22]. There are XML schemas to define contextual information of mobile networks, i.e. device status and reachability. Resource Description Framework (RDF) and Web Ontology Language (OWL) are also widely accepted to represent the context. Composite Capability/Preferences Profile (CC/PP) can be used for capabilities description and user preferences. CC/PP [23] is based on RDF. CC/PP is a widely accepted W3C standard, but it does not specify how contextual information can be stored. Strang and Linnhoff-Popien [24] present a survey of six context modelling approaches: key-value modelling, markup scheme modelling, graphical modelling (UML, ER, etc.), object-oriented modelling, logic-based modelling and ontology-based modelling. According to their analysis that is based on the appropriateness requirements presented in the same chapter, they found that ontology-based modelling is the most promising approach for context modelling in pervasive computing environments.

Troung and Dustdar [25] shed light on context-processing systems designed especially for web services. The Akogrimo project [26] aims at supporting mobile users to access context-aware data on the grid. It focuses on the user's presence, his location and environmental information. It has a context manager that collects contextual information and delivers it to the application. Han et al. [27] present the Anyserver platform, which supports context awareness in mobile web services. This platform utilizes device information, networks and application type as context information, but it does not support context sharing. CoWSAMI [28] also provides a context manager to manage context sources. The ESCAPE [29] framework is a web-based context management system for teamwork and disaster management. It provides web services-based context sensing and sharing front-end as mobile device interface and provides a back-end service for storing and sharing context information. Omnipresent [30] is a location-based service system.

There are currently no context-processing systems aimed at context awareness specifically for mobile applications based on cloud services [25]. These systems aim at few attributes of the context and supporting components of the context processing. Mobility, context storage and context reasoning are not catered in most of the aforementioned systems.

Context awareness is an enabler for service adaptation. Service adaptation is also a complex process, which involves the rapid analysis of the varying context data and the customization of the appropriate services. There are many researchers who presented service adaptation approaches on the basis of the context. Maamar et al. [31] present an approach for service selection and task adaptation for context-based web service composition. They present a CC/PP-aware client proxy, context manager and policies for the personalization of web services dynamically on the basis of the resource context. The policies are used for consistency, feasibility and inspection. In mobile web services, content adaptation is also a common practice. Context information is used to adapt the content for a client request.

Seyler and Taconet [32] present a service orchestration adaptation technique by taking into account the service composition meta-model and context meta-model. Based on these meta-models, they present both deployment time and runtime adaptations. Soukkarieh and his colleague [33] present an architecture aiming at adapting content and presentation of services to the user's context. Their context management module is not asynchronous. User updates its context requirements and then context management updates itself.

Badidi and Esmahi [34] discuss a Software-as-a-Service (SaaS) approach for context information provisioning in the cloud-computing paradigm. They propose a framework for context information provisioning, which relies on deploying context-supporting components in the form of services on the cloud and using context brokers to mediate between context consumers and context services using a publish/subscribe (pub/sub) model. This work is aligned with our approach in a way that we also focus on loosely coupled context-supporting components hosted in the cloud for general-purpose context-aware mobile applications.

8.3 Challenges of Shifting Context Awareness to the Cloud

This section highlights the challenges of developing and running context-aware mobile applications in the cloud, as well as challenges to utilize context awareness in a federated cloud environment. We have distilled these challenges from analysing various context-aware mobile applications [35–37] and what the effects would be for a (partial) deployment in the cloud.

8.3.1 Resource Limitations and Trade-Offs for Advanced Context Processing on Mobile Devices

Mobile cloud computing [38] is a model for transparent elastic augmentation of mobile device capabilities via ubiquitous wireless access to the cloud storage and computing resources. The adjustment of context-aware dynamic off-loading is done with respect to changing operating conditions while preserving the available sensing and interactivity capabilities of the mobile devices. Mobile devices have built-in sensors to sense the real-time situation of the users. As such, they are important for fetching the context data, but the size of the acquired context data varies, depending on the application's objectives. This amount of data can be significant, e.g. the triaxial accelerometer of smartphone used for activity recognition such as step counting or fall detection can generate 100 samples per second, which would produce about 200 kB per minute (one value of type double for each x, y, z axis and a time stamp, producing 32 bytes per sample). Sending all this data to the cloud for statistical feature extraction and selection may be too inconvenient.

Baldauf [14] stressed the importance of the method of context data acquisition when designing context-aware applications because it predefines the architectural style of the application. Most research into context-aware mobile applications has considered the use of various sensors, including position sensors, accelerometers, gyroscopes and cameras. Often, it is not the position itself that constitutes immediately useful context but additional information that can be inferred from the location, for instance, a user in a meeting room implies that he is in a meeting. Accelerometers and gyroscopes provide movement and orientation information. Cameras similarly provide access to potentially rich information that can be derived by means of feature extraction and video analysis for surveillance and assisted living scenarios.

While positional context, motion context and visual context are powerful for augmentation of devices with some awareness of their situation, they also have distinct shortcomings. Position is a static description of an environment and does not capture dynamic aspects of a situation. Its usefulness as context also largely depends on pre-captured knowledge about locations. Motion and vision, on the other hand, can be employed to capture activity and other dynamic aspects, but extraction of a specific context is computationally expensive and problematic in mobile and uncontrolled environments due to the shortage of resources for computation, data storage, network bandwidth and battery capacity. In these cases, context storage and context reasoning are often too impractical to be put on a mobile device. Applications like face detection in media or live gaming [39] can be impossible to run efficiently on a mobile device as it is extremely difficult to manage high-volume data or advanced computational capabilities due to limited resources. Some of the algorithms used for context-aware applications, such as speech recognition or image processing, also require intensive computation and produce large amount of data. Further, it is unreasonable to keep specific types of data on personalized devices, e.g. book ISBN numbers or postcodes of a county. This data is too large and storing it on portable devices is not feasible.

Contrary to mobile devices, cloud computing provides plentiful storage and processing capabilities. Mobile applications are being built on web standards where computation is offloaded to the powerful cloud resources. However, sometimes (such as in the step counting or fall detection cases using the accelerometer on the smartphone) processing mobile data in place on the device would be more efficient and less susceptible to network limitations, compared to off-loading data and processing to the remote cloud.

8.3.2 Connectivity, Latency and Bandwidth

When consuming context-sensitive cloud services on the mobile device, a continuous data communication is required. In a mobile environment, the network connectivity varies from high bandwidth to low bandwidth according to the available network connection. Furthermore, data integration with remote processing requires the local data to be moved to a remote location with the upload speed being highly connection oriented. Hence, even if the processing in the cloud is far more efficient compared to the mobile, the latency triggered by uploading the data can be a challenge to the context-aware cloud services running in the mobile [40], especially if this latency is subject to variations in bandwidth.

Furthermore, what if the connection is lost? A context-aware mobile application completely running in the cloud would stop working if the mobile device loses its connection. Fortunately, new programming languages such as HTML 5 enable data caching locally on the mobile device [41], so the interruptions in connectivity will not affect the ability to continue its operations. However, if the application relies on the cloud for all the context management tasks, the support for its smart behaviour is no longer available.

8.3.3 Security and Privacy

Storing and using the context data, for adapting cloud services, provided by third-party service providers, is a frequently discussed security issue in cloud computing [42]. Clearly, the infrastructure needs to be secure against unauthorized access, but it also needs ways to let people introspect, so that they can understand how their context data is being used. Furthermore, the infrastructure needs to be designed such that privacy concerns are legitimately and adequately addressed. Storing the sensitive information on the client device but allowing preprocessed and encrypted context to be sent to the cloud can address these challenges. But this procedure can be expensive for the mobile device in terms of computation and communication. Imagine a user aims to off-load the entire context processing to the cloud to reduce the computational overhead on his device. In such a scenario, enforcing encryption could really sacrifice any resource gains on the mobile.

8.3.4 Apportioning Decision for Federated Deployments

The biggest challenge for federated deployment of distributed context-aware mobile applications in the cloud is to decide when and what to off-load in an autonomic manner [40]. The context-aware middleware has to take this decision of how to split responsibilities between devices, applications and the middleware itself. It should be able to decide for which use cases the cloud is better and for which ones a cloud-based deployment does not bring any added value. For instance, Hinckley et al. [43] modified a PDA to have tilt, touch and proximity sensors. The screen would rotate simply by rotating the PDA. Also, the voice recorder would automatically activate if the PDA was tilted upwards, was being touched and was near something (ideally the user's face). All of this sensor data was processed locally on the device itself. However, if it were processed in a remote context middleware running in the cloud [42], it is likely that the interactivity would be affected due to network latencies. The essence of the problem is finding the middle ground, determining what the mobile devices and applications should handle and what the cloud should handle.

8.3.5 Managing the Context-Aware Service Life Cycle

Platform-as-a-Service (PaaS) aims to abstract the complexity of managing the life cycle of the cloud services by providing targeted tools and services. The state-of-the-art PaaS offers common services like authentication, authorization, billing and provisioning for multiple Software-as-a-Service (SaaS) applications. Another major challenge for context-aware cloud service providers is to exploit the benefits of cloud computing to manage quality of service (QoS) commitments to customers throughout the life cycle of a service. Context plays a vital role in a service life cycle from service request to service invocation [44], its deployment or customization according to the user's situation and ongoing activities. Context-aware service provisioning and adaptation is technically a challenging job due to continuous varying context and heterogeneous kind of data.

8.4 Architectural Aspects for Context-Aware Mobile Applications in the Cloud

This section presents the architectural considerations and requirements to develop context-aware cloud services. A blueprint of a federated middleware in Platform-as-a-Service layer has been presented with a service-provisioning module to better manage the federation of mobile and cloud platform.

8.4.1 Requirements for Context Processing Within PaaS

The architecture for context-aware mobile applications is evolving to an enormous scale. It is evolving to a many-to-many fashion, where a large number of users can utilize the distributed context-processing middleware asynchronously and this for a great number of personalized services hosted by third-party service providers. Let us exemplify the use of context-based mobile cloud services by a case study of Dr. Peeters and his interaction with the technology:

> Dr. Peeters utilizes his mobile device to watch the morning news redirected on a nearby big screen or only to listen to it by text to speech conversion service, according to the available bandwidth of wireless connection. As he gets into his car to go to work, his smart device is re-configured to low bandwidth wireless connectivity and his desired information is pre-fetched or cached with the help of a context service in the cloud. While travelling, he is interested to see the information of patients who are in his appointment list today. If he has lectures or meetings, he would like to get notified. His device can offer him a text to speech conversion for his agenda. He can stream his online music repository to his mobile device as well. He is not worried on being late as he knows traffic is smooth as otherwise his intelligent mobile device would have notified him with an alternative route. During the day, Dr. Peeters manages multiple patients' cases simultaneously on the basis of their emergency of which he gets informed through push notifications. Other cloud services enable him to collaborate with his peers effectively and to participate in discussions remotely.

In the above-mentioned use case scenario, we cannot neglect the role of context information, as it is vital to make all the services work in harmony and adapt dynamically to achieve runtime intelligent behaviour of the services in a scalable way. The enabling factors for these interconnected mobile services are context-aware intelligence, interoperability, standardized communication protocols and platforms to deploy and efficiently adapt these services. Any software architecture designed for context-aware mobile applications, federated with the cloud services, will need to foresee the increasing heterogeneity of context-capturing devices. It will also have to provide service provisioning for varying context.

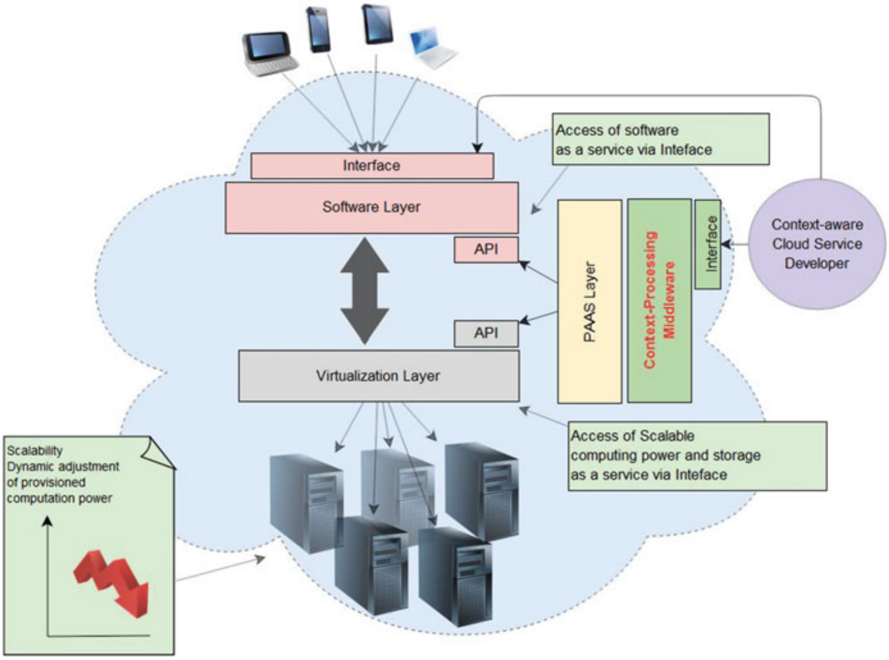

Fig. 8.1 Context processing in the cloud-computing model

In our opinion, there is a strong need of a web service-based context-awareness abstraction as a part of the PaaS layer in the cloud. Figure 8.1 shows how context awareness affects the cloud-computing model and interaction of a context-aware cloud service developer with the layers of cloud-computing paradigm. A context-processing middleware is shown as an abstracted layer within Platform-as-a-Service to develop context-aware cloud services and their dynamic adaptation. Platform-as-a-Service is already providing service provision for scalability of cloud services. The context-processing middleware can utilize the same service-provisioning infrastructure with the addition of a few more services to manage and adapt services on the basis of the varying context.

It would ease the context-aware application development, the context processing and service provisioning at runtime. The biggest benefit of such a type of middleware is that the sensors, services and devices can both be changed independently, even dynamically when other sensors, services, devices and applications are running. The encapsulated context processing increases the reusability and eases the development of large-scale, smart mobile applications.

We summarize below the architectural requirements for a context-processing middleware, as a part of the Platform-as-a-Service layer:

- General-purpose cloud services must not be tightly coupled to the context-processing middleware. It allows the context management to evolve as new sources of information become available or existing sources disappear.

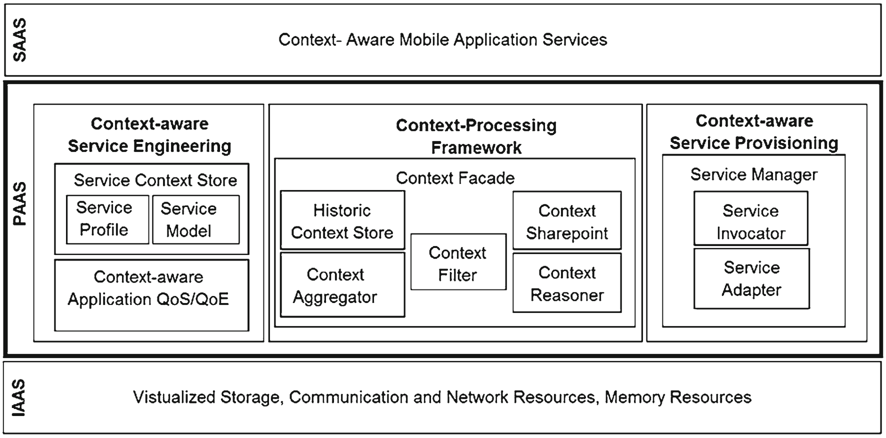
SAAS
Context- Aware Mobile Application Services
PAAS
Context-aware Service Engineering
Service Context Store
Service Profile
Service Model
Context-aware Application QoS/QoE
Context-Processing Framework
Context Facade
Historic Context Store
Context Filter
Context Sharepoint
Context Aggregator
Context Reasoner
Context-aware Service Provisioning
Service Manager
Service Invocator
Service Adapter
IAAS
Vistualized Storage, Communication and Network Resources, Memory Resources

Fig. 8.2 Architecture of the context-processing middleware and context-aware service provisioning in the PaaS

- A separate abstraction layer of a service manager is required to match the context of the user and the services that the user has subscribe to.
- The service manager requires advanced adaptation algorithms to adapt cloud services according to the evolving context of the user and his device.
- A trade-off analysis technique is required for the apportioning decision of how to dynamically infer what and when to off-load to the cloud without affecting the QoS and QoE.
- Users should be able to add their own preferences and requirements about which context information and applications should remain local on the mobile.
- The architecture needs to be open and extensible to integrate new context management components, while at the same time remain scalable and elastic to handle a growing number of users.

Figure 8.2 depicts the three major building blocks of the context-aware cloud services in the PaaS

8.4.2 Information Viewpoint of the Context-Processing Architecture

In context-aware applications, the context data is highly distributed and possibly coming from and being used anywhere, anytime. Smart applications based on context-aware cloud services differ from traditional applications because the information processing life cycle is highly distributed. It is important that the interpretation of the context information is uniform for every participating cloud service. We leveraged our context ontology [45] to describe the contextual concepts as well as

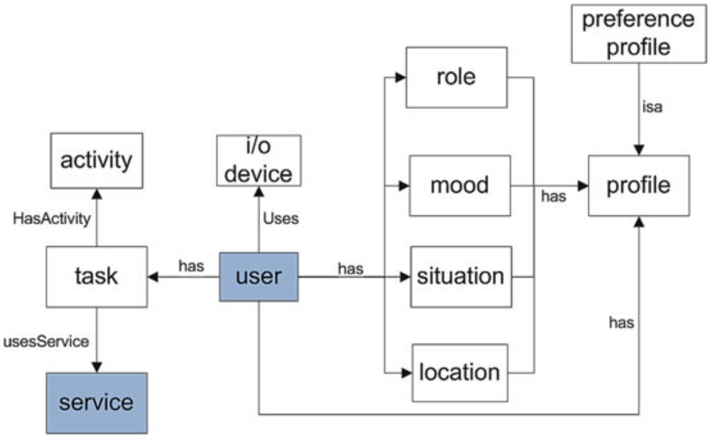

Fig. 8.3 User context ontology [46]

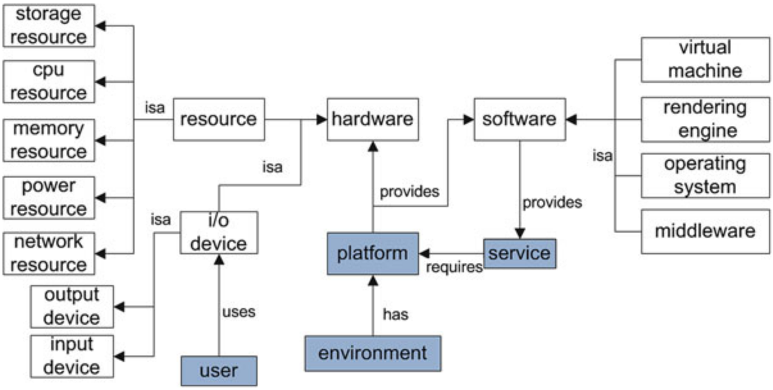

Fig. 8.4 Platform context ontology [46]

their interrelationships. The three major context concepts in our ontology [46] that we use in our architectural information viewpoint are the following.

1. *User context*: Any information related to user of the cloud service is regarded as user context. It can be location, activities, habits, preferences, noise level, light, temperature, etc. Figure 8.3 shows the ontology design for user's context.
2. *Platform context*: The information related to the mobile device that is significant for the execution of a service is classified as device context, e.g. battery level, available bandwidth and display resolution (Fig. 8.4).
3. *Service context*: Any information related to the service during its life cycle relevant for service adaptation, e.g. the functionality description, the interfaces and required resources. This type of information is essential for the service manager (see Fig. 8.2) to take the apportioning decision. Figure 8.5 shows the context of a service.

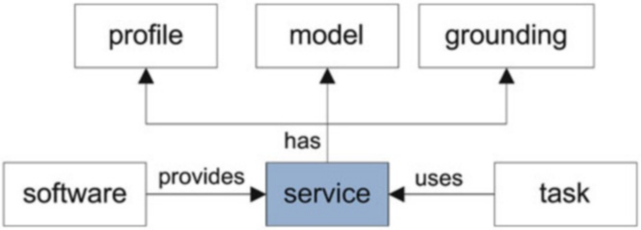

Fig. 8.5 Service context ontology [46]

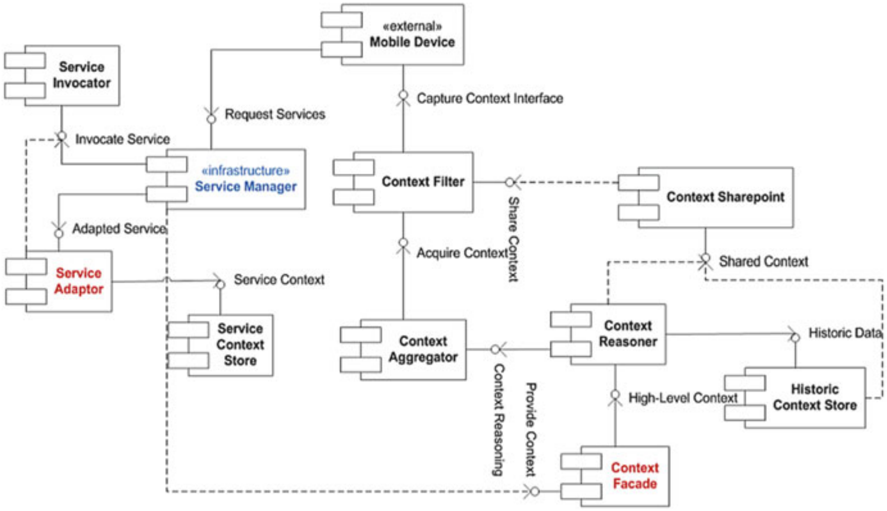

Fig. 8.6 Functional viewpoint of the architecture of context-processing middleware

8.4.3 Functional Viewpoint of the Context-Processing Architecture

Context awareness enables automatic service adaptation. It helps the system to adapt its behaviour according to desired objectives while offering interoperable and scalable personalized services. Figure 8.6 presents the functional viewpoint of the architectural design presented in Fig. 8.2 in the PaaS layer. This functional view of the architecture is designed to cater the functional requirements of a context-processing middleware. It defines the key functional elements, their responsibilities, the interfaces they expose and the interaction between them. In PaaS, there is already a mechanism to provision web services and on the fly creation of new services that do not disturb the existing fabric. However, we need a service manager for context-aware adaptation of the services. The service manager will take into account the context of users, devices and resources available to adapt to the best matching service as discussed in Sect. 8.4.2.

A mobile device requests a service to the *Service Manager* in the cloud. If the service is not context aware, the *Service Manager* invokes the service with the coordination of the *Service Invocator*. If the service is context aware, the *Service Manager* contacts the context-processing middleware component called *Context Façade*. *Context Façade* acts as an abstraction to other context-processing components. This service pushes the request of context information to the *Context Reasoner*. The context reasoning service asks the *Context Aggregator* and *Historic Context Store* for preprocessed and historic context data. It infers the requested context and provides it to the *Context Façade*. The *Context Sharepoint* service, to acquire shared context, is shown in this diagram as an optional service. The *Context Sharepoint* service contacts the *Context Filter* service to provide context of multiple users. Once the context of the user is determined by the *Context Façade*, the *Service Manager* accesses the context and contacts the *Service Adapter* to invoke the required service compatible to the current context. The *Service Adapter* is dependent on the *Service Context Store*. *Service Context Store* is a place where semantically defined services are published and each service has a valid context profile. The *Service Adapter* matches the service context profile with the user context and customizes the service accordingly. The *Service Manager* calls the *Service Adapter* service whenever the context changes.

8.5 Implementation and Evaluation

We have recognized a few specialized services on the basis of the use case scenario from Sect. 8.4.1. There are interlinked services running on a mobile device. We have distributed the context-awareness process in nine autonomous services. Figure 8.7 shows the high-level composition and interaction of these services.

We are using WSO2 Stratos [47], an open PaaS solution, for implementing our context-aware components as loosely coupled services and our empirical evaluation. We are in the process of evaluating the scalability of the context-processing middleware for a context-aware mobile healthcare application [48] using human activity recognition as a composition of several context-processing services. Although we have preliminary results of small-scale experiments, they do not provide any statistical significance for large-scale deployments with many users. For that, we need large context datasets and business-level ecosystems to evaluate different provisioning strategies in order to provide evidence regarding optimal strategies that improve customer QoE. Once our context-processing middleware is fully operational, we will also cross-validate the performance of our adaptation strategies by running the same instrumented applications on existing public cloud infrastructures (e.g. Amazon or Google App Engine).

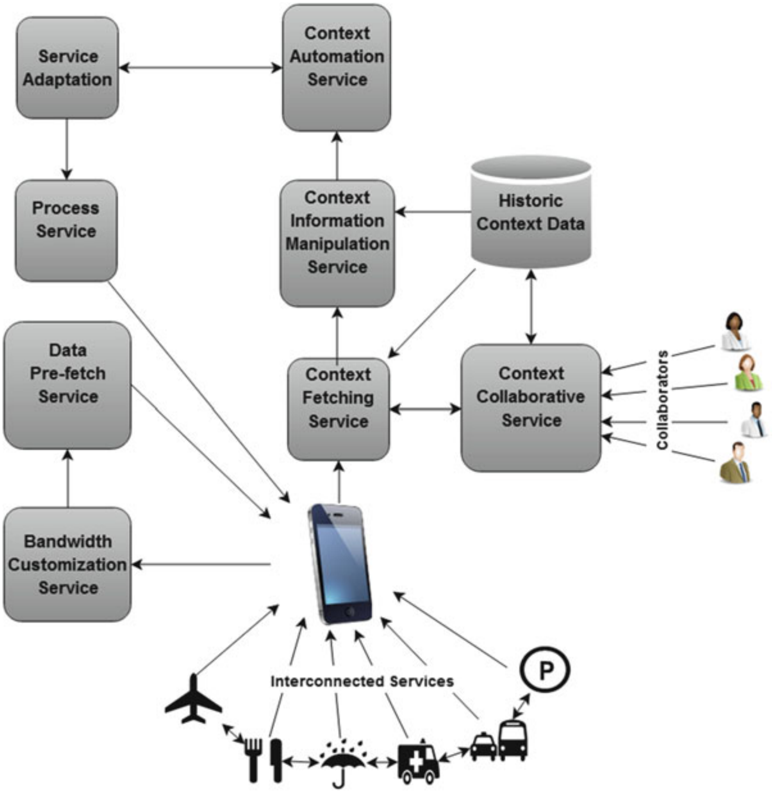

Fig. 8.7 Composition of the context-aware mobile services

8.6 A Research Road Map for Future Context-Aware Mobile Cloud Services

Cloud computing is a building block of the Future Internet, and context awareness is an integrated part of the Future Internet. This enables a common vision for the deployment of independent federated services and applications, characterized by a high degree of autonomous data capture, event transfer, network connectivity and interoperability. The added value is in context-aware mobile applications, such as environmental monitoring and healthcare monitoring [48], where the high volumes and rates of data need rapid processing to translate raw data into contextualized information. The context-aware self-adapting mobile applications will support the interaction between heterogeneous context-aware cloud services.

The context-processing and context-aware service provisioning in PaaS will give rise to the interactions among context-aware cloud services in order to be able to support heterogeneous sources of context data and many heterogeneous mobile devices through the use of standard interfaces and data models to ensure a high degree of interoperability among diverse context-aware applications.

From an architectural perspective, we need adoption of standards to promote interoperability and to allow each context-aware service to benefit from a competitive marketplace of interoperable technology solutions from multiple service providers. As greater trust is placed on the context-processing and service-provisioning middleware in PaaS, it will be essential to ensure that international quality and integrity standards are deployed and further developed. Beyond the need for standardization to ensure interoperability, this requires further research in the following areas: (1) ontology-based semantic standards for context-aware cloud services, (2) protocols for communication within and outside cloud environments, (3) efficient and advanced service management and adaptation algorithms and (4) ways to effectively apportion the context-processing services within mobile and cloud platforms. We strongly believe that these areas are part of the road map for future research towards mobile cloud computing with a context-awareness perspective.

8.7 Conclusion

Context-based cloud services and mobile connectivity will give rise to the new interaction with the cloud in which information and communication is embedded in the environment around us in the form of wireless networks. As a result of the extraordinary increase of modern mobile Internet-enabled devices and the pervasive wireless and cellular data networks, a large number of mobile users are requiring personalization services customized to their context. The next wave in the era of cloud computing will be outside the realm of the traditional cloud services and infrastructure. Most of the cloud services will be context aware.

In this chapter, we focused on the need of loosely coupled context-supporting components that work with a context-aware service-provisioning infrastructure to adapt services to the context of the user and his mobile device. We believe that the adoption of our PaaS-based context-processing middleware will allow future mobile cloud solutions to be developed quicker by abstracting all context management into reusable services, as well as accelerate research in the area of context-aware mobile applications for the cloud.

Acknowledgments This research is partially funded by the Interuniversity Attraction Poles Programme Belgian State, Belgian Science Policy and by the Research Fund KU Leuven.

References

1. Buyya, R., Chee Shin, Y., Venugopal, S.: Market-oriented cloud computing: vision, hype, and reality for delivering IT services as computing utilities. In: 10th IEEE International Conference on High Performance Computing and Communications, 2008, 25–27 Sept 2008, pp. 5–13. Dalian, China, (2008)
2. Christensen, J.H.: Using RESTful web-services and cloud computing to create next generation mobile applications. In: OOPSLA '09, 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems Languages and Applications, 2009, pp 627–634. ACM, New York. doi:10.1145/1639950.1639958
3. Papakos, P., Capra, L., Rosenblum, D.S.: VOLARE: context-aware adaptive cloud service discovery for mobile systems. Paper presented at the Proceedings of the 9th international workshop on adaptive and reflective middleware, Bangalore, 2010
4. Dey, A.K.: Understanding and using context. Pers. Ubiquit. Comput. **5**(1), 125–134 (2001)
5. Saha, D., Mukherjee, A.: Pervasive computing: a paradigm for the 21st century. Computer **36**(3), 25–31 (2003). doi:10.1109/mc.2003.1185214
6. Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M., Steggles, P.: Towards a better understanding of context and context-awareness. In: International Symposium on Handheld and Ubiquitous Computing, Karlsruhe, Germany, pp. 304–307. Springer, London (1999)
7. Schilit, B.N., Theimer, M.M.: Disseminating active map information to mobile hosts. IEEE Netw. **8**(5), 22–32 (1994). doi:10.1109/65.313011
8. Chen, G., Kotz, D.: A survey of context-aware mobile computing research. Dartmouth college, Technical report. Dartmouth college, Hanover (2000)
9. Chaari, T., Ejigu, D., Laforest, F., Scuturici, V.-M.: A comprehensive approach to model and use context for adapting applications in pervasive environments. J. Syst. Softw. **80**(12), 1973–1992 (2007). doi:10.1016/j.jss.2007.03.010
10. Gellersen, H.W., Schmidt, A., Beigl, M.: Multi-sensor context-awareness in mobile devices and smart artifacts. Mob. Netw. Appl. **7**(5), 341–351 (2002)
11. Dey, A.K., Häkkilä, J.: Context-awareness and mobile devices. In: Handbook of Research on User Interface Design and Evaluation for Mobile Technology, vol. 2, pp. 205–217. Information Science Reference/IGI Global, New York/Hershey (2008)
12. Chalmers, D., Dulay, N., Sloman, M.: A framework for contextual mediation in mobile and ubiquitous computing applied to the context-aware adaptation of maps. Pers. Ubiquit. Comput. **8**(1), 1–18 (2004)
13. Kim, Y.S., Lee, K.H.: A lightweight framework for mobile web services. Comput. Sci. Res. Dev. **24**(4), 199–209 (2009)
14. Baldauf, M.: A survey on context-aware systems. Ad Hoc Ubiq. Comput. **2**(4), 263–277 (2007)
15. JoonSeok, P., Mikyeong, M., Seongjin, H., Keunhyuk, Y.: CASS: a context-aware simulation system for smart home. In: 5th ACIS International Conference on Software Engineering Research, Management & Applications, 2007, 20–22 Aug. 2007. pp. 461–467. Busan, South Korea (2007). doi:10.1109/sera.2007.60
16. Tao, G., Pung, H.K., Zhang, D.Q.: A middleware for building context-aware mobile services. In: IEEE 59th, Vehicular Technology Conference, 2004, 17–19 May 2004, vol. 2655, pp. 2656–2660. Milan, Italy (2004). doi:10.1109/vetecs.2004.1391402
17. Chen, H., Finin, T., Joshi, A.: Semantic Web in the context broker architecture. In: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications, 2004, 14–17 March 2004, pp. 277–286. Orlando, Florida (2004). doi:10.1109/percom.2004.1276865
18. Hofer, T., Schwinger, W., Pichler, M., Leonhartsberger, G., Altmann, J., Retschitzegger, W.: Context-awareness on mobile devices – the hydrogen approach. In: Paper Presented at the

Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03), Hawaii, 10 pp. (2003)

19. Salber, D., Dey, A.K., Abowd, G.D.: The context toolkit: aiding the development of context-enabled applications. Paper presented at the Proceedings of the SIGCHI conference on Human Factors in Computing Systems: The CHI Is the Limit, Pittsburgh (1999)
20. Hong, J.I., Landay, J.A.: An infrastructure approach to context-aware computing. Hum. Comput. Interact. **16**(2), 287–303 (2001)
21. Mell, P., Granc, T.: The NIST Definition of Cloud Computing [NIST special publication]. National Institute of Standards and Technology, Gaithersburg, Maryland, USA (2011)
22. Ferschas, A., Hechinger, M., Riener, A., Schmitzberger, H., Franz, M., Rocha, M.S., Zeidler, A.: Context-aware profiles. In: International Conference on Autonomic and Autonomous Systems, Silicon Valley, California, USA. 2006, pp. 48–48. IEEE (2006)
23. Butler, M.H.: Current Technologies for Device Independence. HP Laboratories Bristol, UK (technical report) 83 (2001)
24. Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., Riboni, D.: A survey of context modelling and reasoning techniques. Pervasive Mob. Comput. **6**(2), 161–180 (2010). doi:10.1016/j.pmcj.2009.06.002
25. Truong, H.L., Dustdar, S.: A survey on context-aware web service systems. Int. J. Web Inf. Syst. **5**(1), 5–31 (2009)
26. Jähnert, J., Villagrá, V., Wesner, S.: The Akogrimo Mobile Grid Reference Architecture. Akogrimo Whitepaper. http://www.mobilegrids.org (2006)
27. Han, B., Jia, W., Shen, J., Yuen, M.C.: Context-awareness in mobile web services. In: Parallel and Distributed Processing and Applications, pp. 519–528. Springer Berlin Heidelberg (Book Chapter) (2005)
28. Athanasopoulos, D., Zarras, A.V., Issarny, V., Pitoura, E., Vassiliadis, P.: CoWSAMI: interface-aware context gathering in ambient intelligence environments. Pervasive Mob. Comput. **4**(3), 360–389 (2008)
29. Truong, H.L., Juszczyk, L., Manzoor, A., Dustdar, S.: ESCAPE: an adaptive framework for managing and providing context information in emergency situations. In: EuroSSC'07 Proceedings of the 2nd European Conference on Smart Sensing and Context, 2007, pp 207–222. Springer, Berlin/Heidelberg (2007)
30. De Almeida, D.R., de Souza Baptista, C., Da Silva, E.R., Campelo, C.E.C., De Figueiredo, H.F., Lacerda, Y.A.: A context-aware system based on service-oriented architecture. In: 20th International Conference on Advanced Information Networking and Applications. Vienna, Austria 2006
31. Maamar, Z., AlKhatib, G., Mostefaoui, S.K.: Context-based personalization of web services composition and provisioning. In: IEEE 30th Euromicro Conference, 2004, pp. 396–403. Rennes, France (2004)
32. Seyler, F., Taconet, C., Bernard, G.: Context adaptation of web service orchestrations. In: 16th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2007, pp. 351–356. Paris, France (2007)
33. Soukkarieh, B., Sedes, F.: Integrating a context model in web services. In: IEEE International Conference on Web Services, 2007, pp. 1195–1196. Salt lake city, UTAH (2007)
34. Badidi, E., Esmahi, L.: A cloud-based approach for context information provisioning. World Comput. Sci. Inf. Technol. J. **1**(3), 63–70 (2011)
35. McGraw, I., Lee, C., Hetherington, L., Seneff, S., Glass, J.: Collecting voices from the cloud. In: Proceedings of LREC, Malta, May 2010
36. Angin, P., Bhargava, B., Helal, S.: A mobile-cloud collaborative traffic lights detector for blind navigation. In: Eleventh IEEE International Conference on Mobile Data Management, 2010, pp. 396–401. Kanas City, Missouri, USA (2010)
37. Abowd, G.D., Atkeson, C.G., Hong, J., Long, S., Kooper, R., Pinkerton, M.: Cyberguide: a mobile context-aware tour guide. Wirel. Netw. **3**(5), 421–433 (1997)

38. Kovachev, D., Cao, Y., Klamma, R.: Mobile Cloud Computing: A Comparison of Application Models. Arxiv preprint arXiv:11074940 (2011)
39. Chun, B-G., Maniatis, P.: Augmented smartphone applications through clone cloud execution. Paper presented at the Proceedings of the 12th conference on Hot Topics in Operating Systems, Monte Verit. Springer-Verlag Berlin, Heidelberg 2009
40. Kim, J.: Design and evaluation of mobile applications with full and partial offloadings. In: Advances in Grid and Pervasive Computing, pp. 172–182 (2012)
41. Pilgrim, M. (ed.): HTML5: Up and Running, 1st edn. O'Reilly Media, Sebastopol (2010)
42. Kumar, K., Yung-Hsiang, L.: Cloud computing for mobile users: can offloading computation save energy? Computer **43**(4), 51–56 (2010)
43. Hinckley, K., Pierce, J., Sinclair, M., Horvitz, E.: Sensing techniques for mobile interaction. Paper presented at the Proceedings of the 13th Annual ACM Symposium on User Interface software and Technology, San Diego, 2000
44. Sivashanmugam, K., Verma, K., Sheth, A., Miller, J.: Adding semantics to web services standards. In: International Conference on Semantic Web and Web Services, 2003, pp. 395–401. Erfurt, Germany (2003)
45. Preuveneers, D., Berbers, Y.: Adaptive context management using a component-based approach. In: Distributed Applications and Interoperable Systems, Athens, 2005, pp 1080–1082. Springer, Berlin/Heidelberg (2005)
46. Preuveneers, D., Van den Bergh, J., Wagelaar, D., Georges, A., Rigole, P., Clerckx, T., Berbers, Y., Coninx, K., Jonckers, V., De Bosschere, K.: Towards an extensible context ontology for ambient intelligence. In: Ambient Intelligence: Second European Symposium, EUSAI, Eindhoven, 2004, pp. 148–159. ACM, New York (2004)
47. Azeez, A., Perera, S., Weerawarana, S., Fremantle, P., Uthaiyashankar, S., Abesinghe, S.: WSO2 Stratos: an application stack to support cloud computing. it-Inf. Technol. **53**(4), 180–187 (2011)
48. Preuveneers, D., Berbers, Y.: Mobile phones assisting with health self-care: a diabetes case study. In: Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services, 2008, pp 177–186. ACM, New York (2008)